EXOR INTERNATIONAL

Tech-note

CODESYS V2.3 Ethernet Driver for JMobile

This documents describes how to use the CODESYS Ethernet driver for JMobile platform

EXOR International S.p.A.
Ptn0344
Ver. 1.09

EXOR INTERNATIONAL **Tech-note**

Copyright © 2013 EXOR International S.p.A. – Verona, Italy
Subject to change without notice
The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.
Third-party brands and names are the property of their respective owners.
www.uniop.com

Contents

CoDeSys V2.3 Ethernet Driver 5
Protocol Editor Settings 5
CODESYS Software Settings 8
Tag Import 9
Aliasing Tag Names in Network Configurations 10
1.1 Tag Array 11
Data Types 14
Standard Data Types 14
Special Data Types 15
Limitations 16
Communication Status 16

CoDeSys V2.3 Ethernet Driver

The CODESYS communication driver for Ethernet supports communication switch controllers based on the V2.3 CODESYS version.
Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Accordingly, always ensure that the latest driver is used in the application.

Document code ptn0344
Version 1.09

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called “CODESYS V2 ETH” from the list of available protocols.

The CODESYS V2 ETH driver supports three different protocol types:

- Level2
- Level2 Route
- Level4

The protocol type can be selected from the dedicated combo box in the dialog.

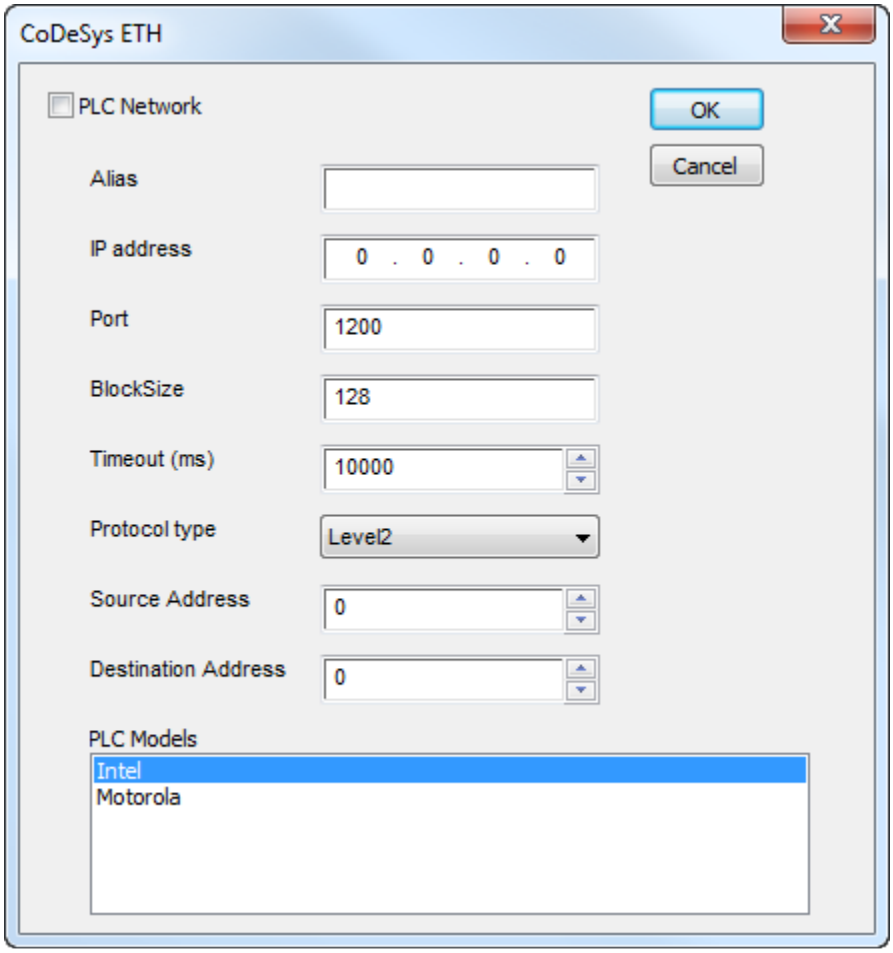


Figure 1

Alias Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node

IP address Ethernet IP address of the controller

Port This parameter allows changing the port number used for the communication. Default value for this parameter is set to 1200 and it corresponds to the default setting of CODESYS -based controllers.

PLC Model Defines the byte order that will be used by the communication driver when sending communication frames to the PLC

Block Size Enter the max block size supported by your controller (limit is 1024 KB)

Protocol type Shows a list with the available protocol variants. Please make sure you check what protocol variant is supported by the CODESYS run-time you want to connect.

Timeout The number of milliseconds between retries when communication fails.

Source Address & Destination Address

Source and Destination Address are available only when TCP/IP Level 2 Route is selected in Controller Setup. The Destination is the node of the PLC and allows the protocol to reads variable in a sub-network. The address is used to read variables when multiple PLCs are connected in a sub-network (serial network) but only one of it have the Ethernet interface.

PLC Network

The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check “Access Multiple PLC’s” checkbox and enter IP Address for all controllers.

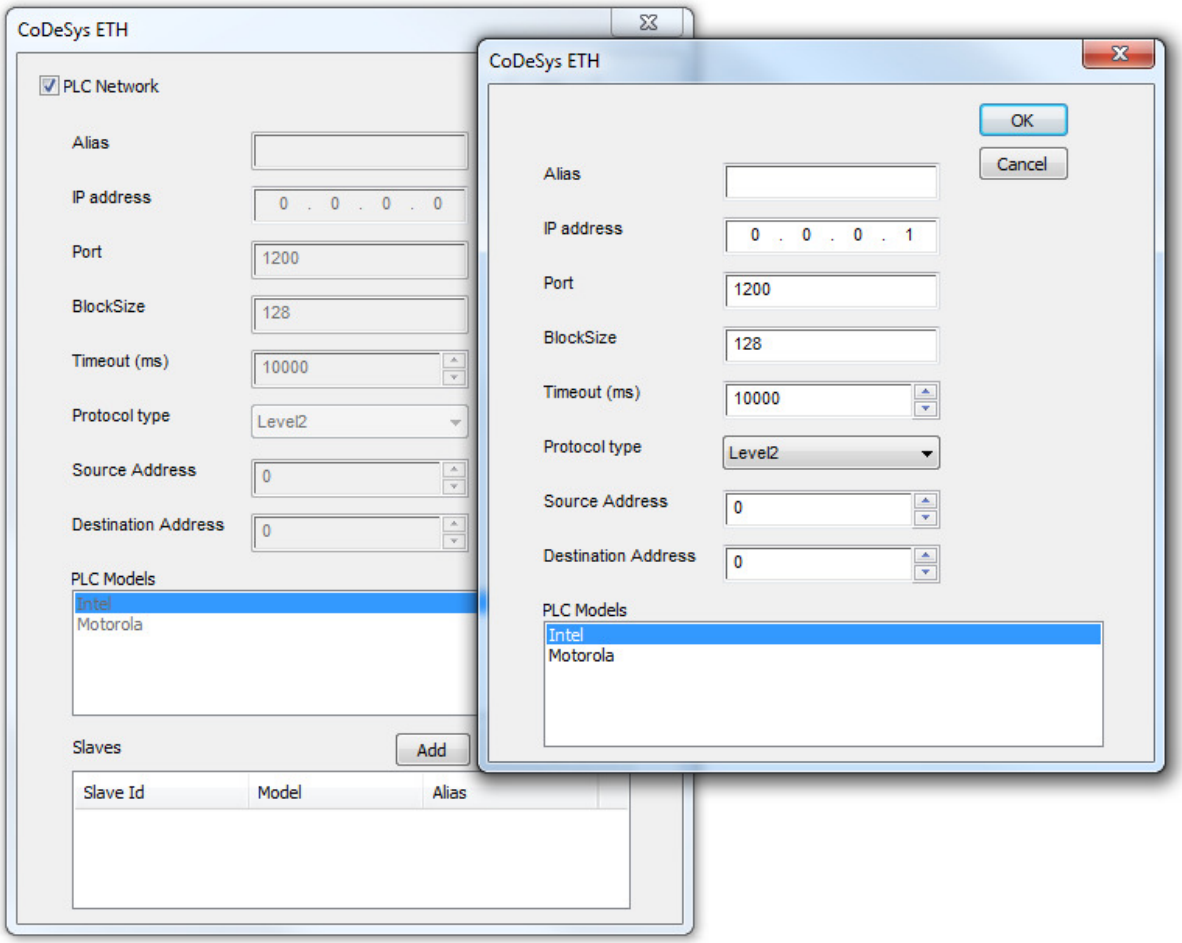


Figure 2

Note: The CODESYS V2 Ethernet driver supports the connections to multiple controllers starting from version V1.60

Note: The CODESYS V2 ETH driver is the best choice also when creating projects for the internal controller iPLC CODESYS. To use the CODESYS V2 ETH driver for communication with iPLC it is enough to configure the IP address of the PLC as localhost (127.0.0.1). The iPLC CODESYS supports communication with CODESYS V2 ETH driver with symbol based support (see next chapter) starting from V1.55 and above.

CODESYS Software Settings

When creating the project in CODESYS, the option Download Symbol File (in Target Settings/General) must be checked.

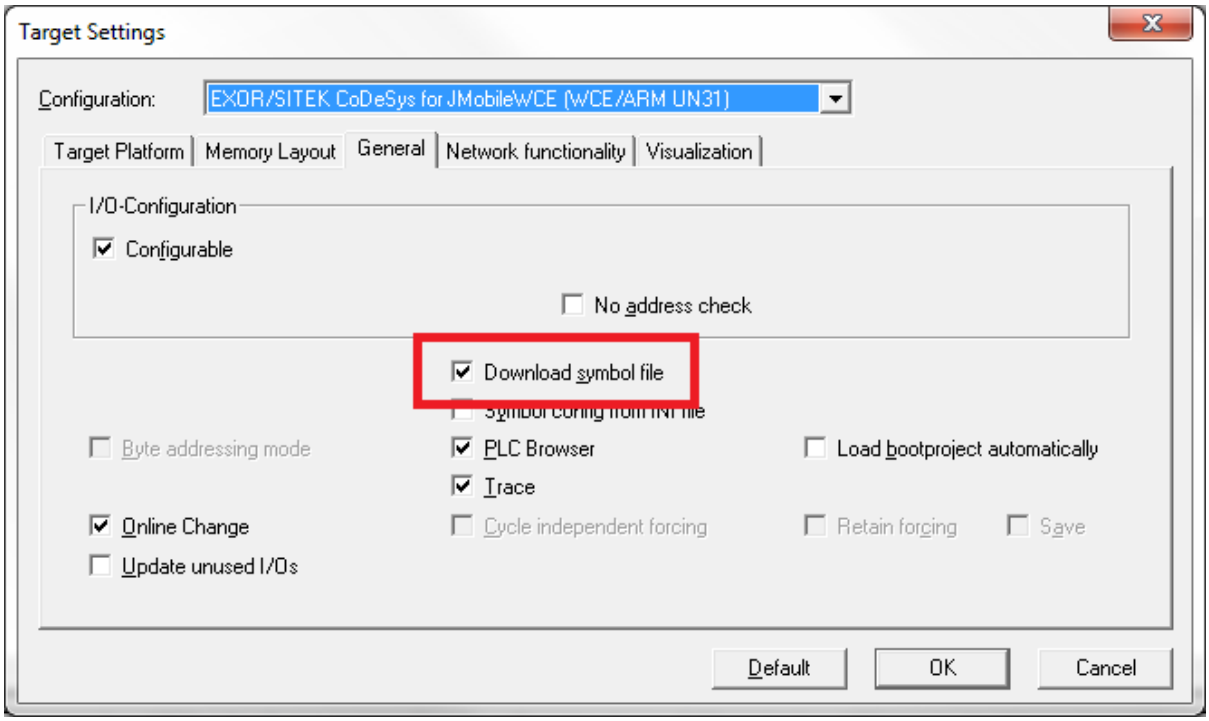


Figure 3

Note: CODESYS V2 ETH communication driver supports the automatic symbol file (SDB) upload from the PLC; any change in the tag offset due to new compilation of the PLC program does not require a symbol file re-import. Tag file has to be re-imported only in case of tag rename or definition of new tags.

When the option Download symbol file is not available or not checked, the protocol can work only if the ProjectId tag is imported. Any change in the tag offset due to new compilation of the PLC program requires that symbol file is imported again.

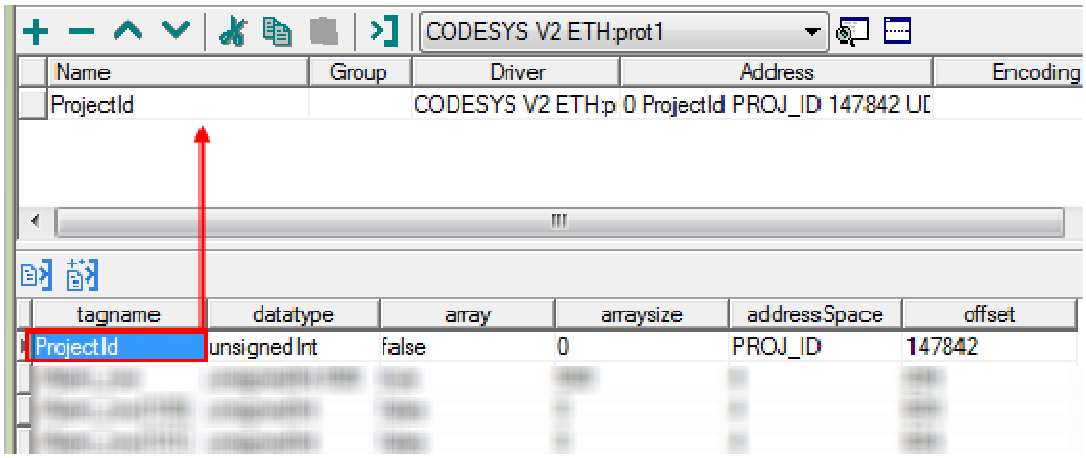


Figure 4

Tag Import

When configuring PLC using the manufacturer's configuration software, make sure to enable Symbol file creation (file with .SYM extension). It can be done under the CODESYS programming software, by selecting "Project\Option\Symbol configuration" and mark the check box "Dump symbol entries" as shown in the picture below.

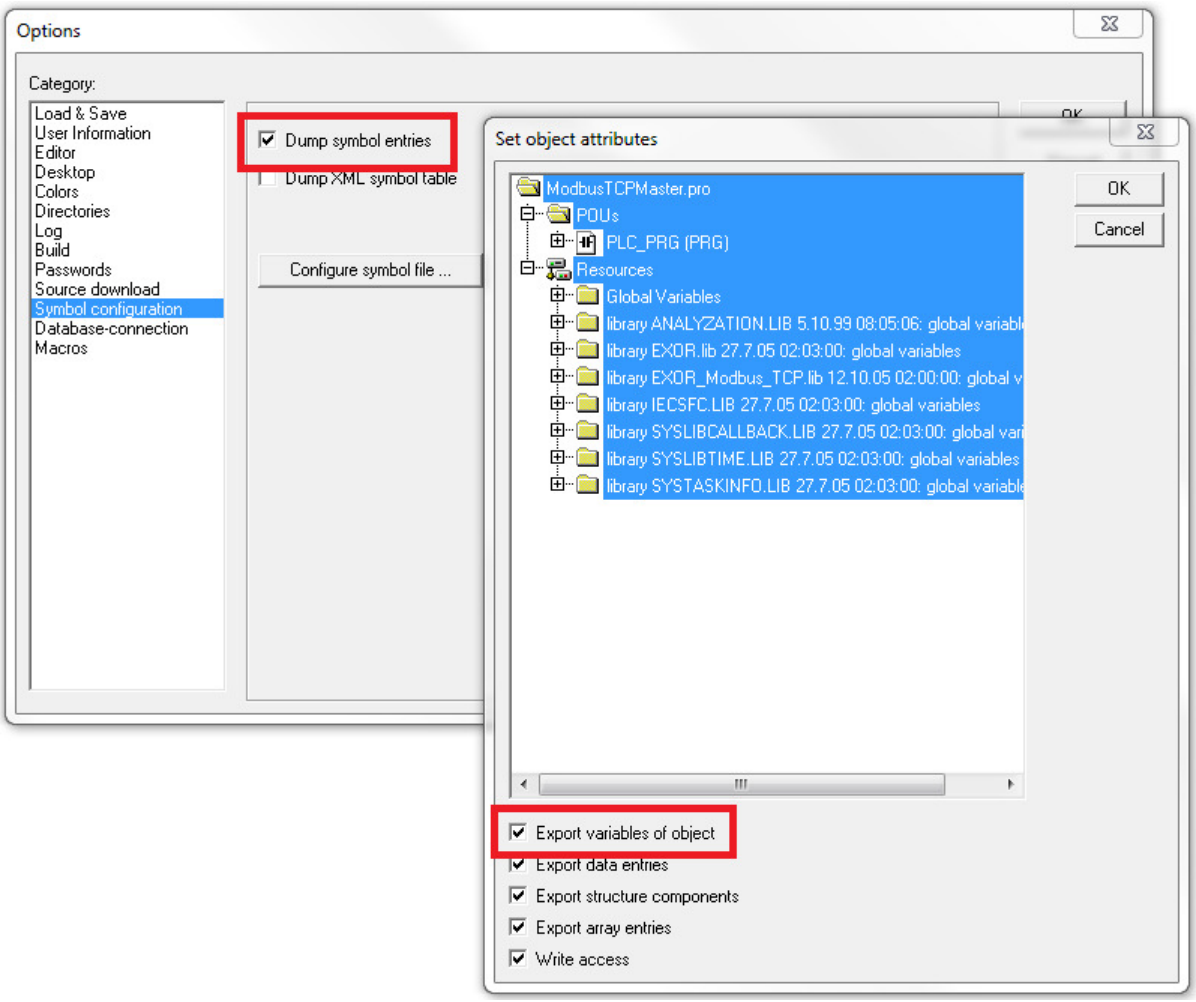


Figure 5

Note: Click then on the "Configure symbol file…" button and make sure the "Export variables of object" check box is marked as shown in the following picture. We recommend to un-check the check box and mark it again to be sure about the proper settings.

Select the driver in the Studio tag editor and click on the "Import tag" button to start the importer.

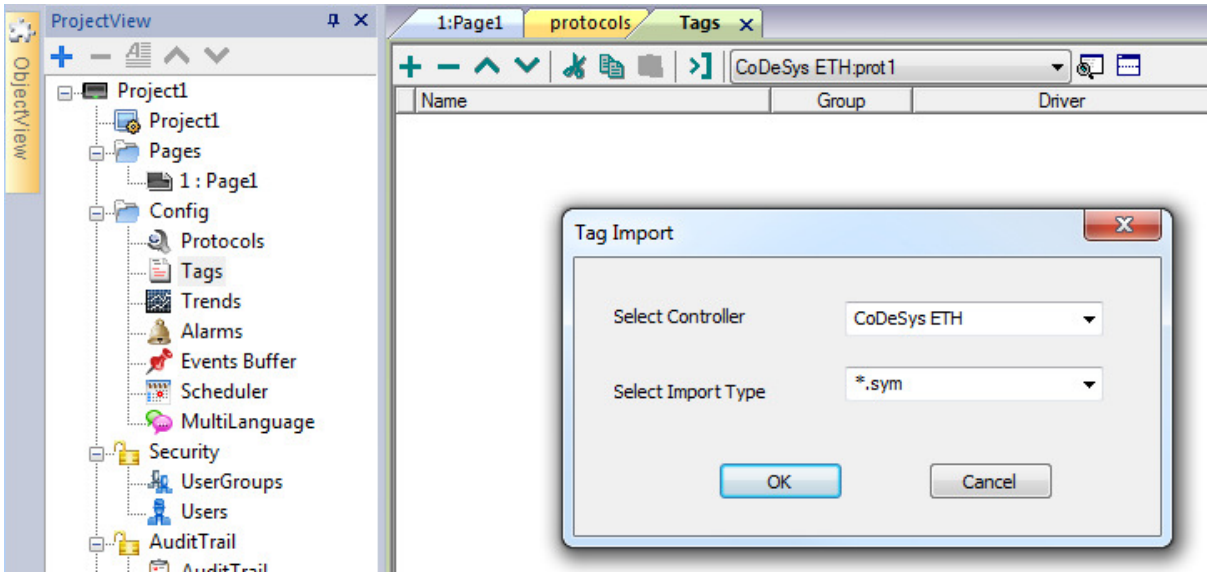
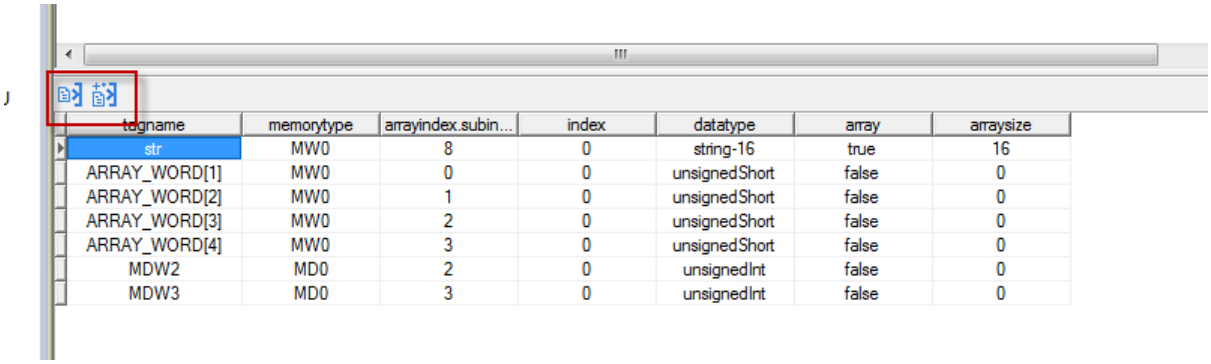


Figure 6

Locate the ".sym" file and confirm.
The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in the following figure.



tagname	memorytype	arrayindex.subin...	index	datatype	array	arraysize
str	MW0	8	0	string-16	true	16
ARRAY_WORD[1]	MW0	0	0	unsignedShort	false	0
ARRAY_WORD[2]	MW0	1	0	unsignedShort	false	0
ARRAY_WORD[3]	MW0	2	0	unsignedShort	false	0
ARRAY_WORD[4]	MW0	3	0	unsignedShort	false	0
MDW2	MD0	2	0	unsignedInt	false	0
MDW3	MD0	3	0	unsignedInt	false	0

Figure 7

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.
The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the "Alias". As shown in the figure below, the connection to a certain controller is assigned the name "Node1". When tags are imported for this node, all tag names will have the prefix "Node1" making each of them unique at the network/project level.

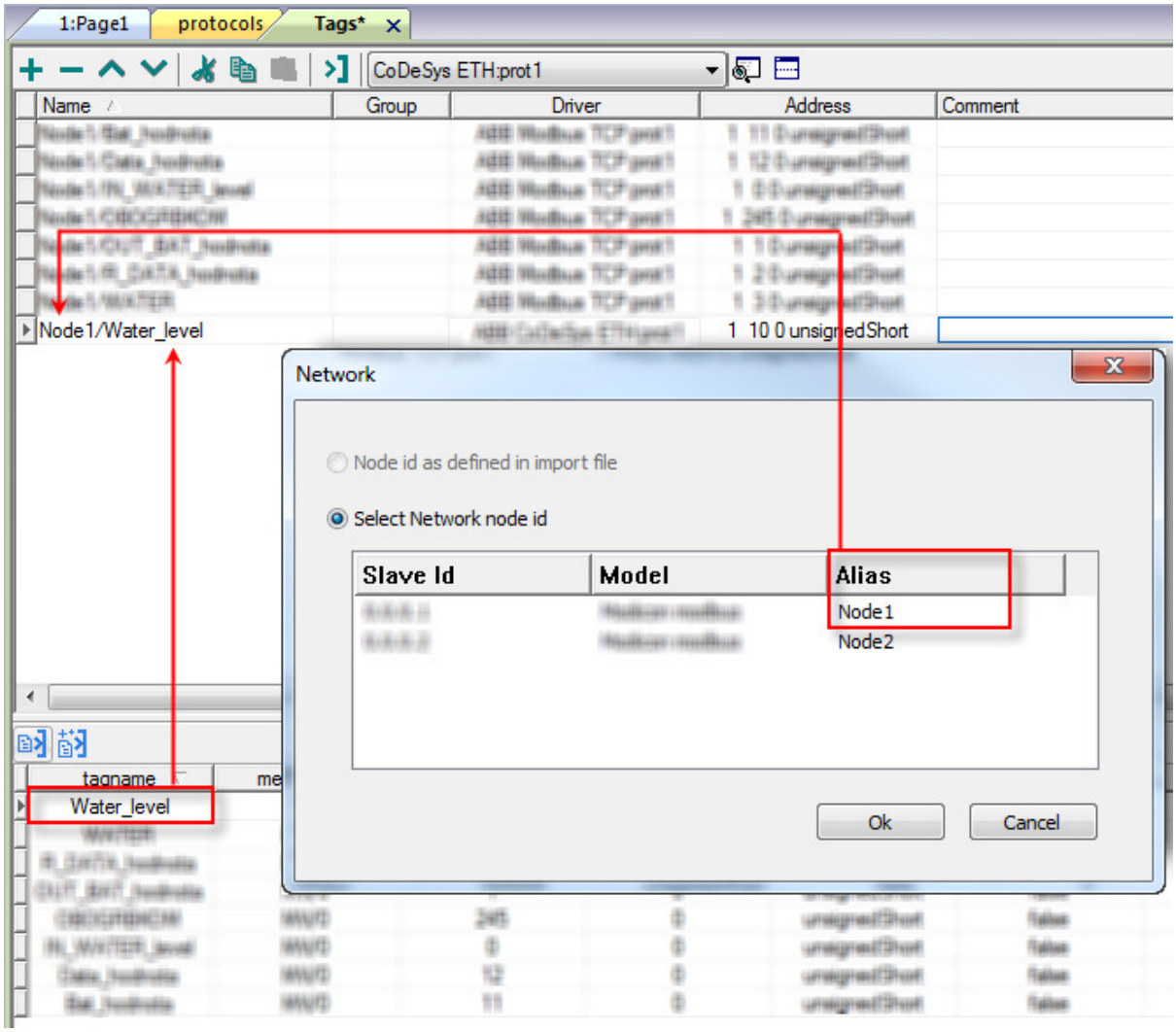


Figure 8

Note:
An Aliasing tag name is only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name. The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

1.1 Tag Array

Tag Arrays are split into individual elements and one Tag for each element is created. The figure below shows an example of one Array with 10 elements

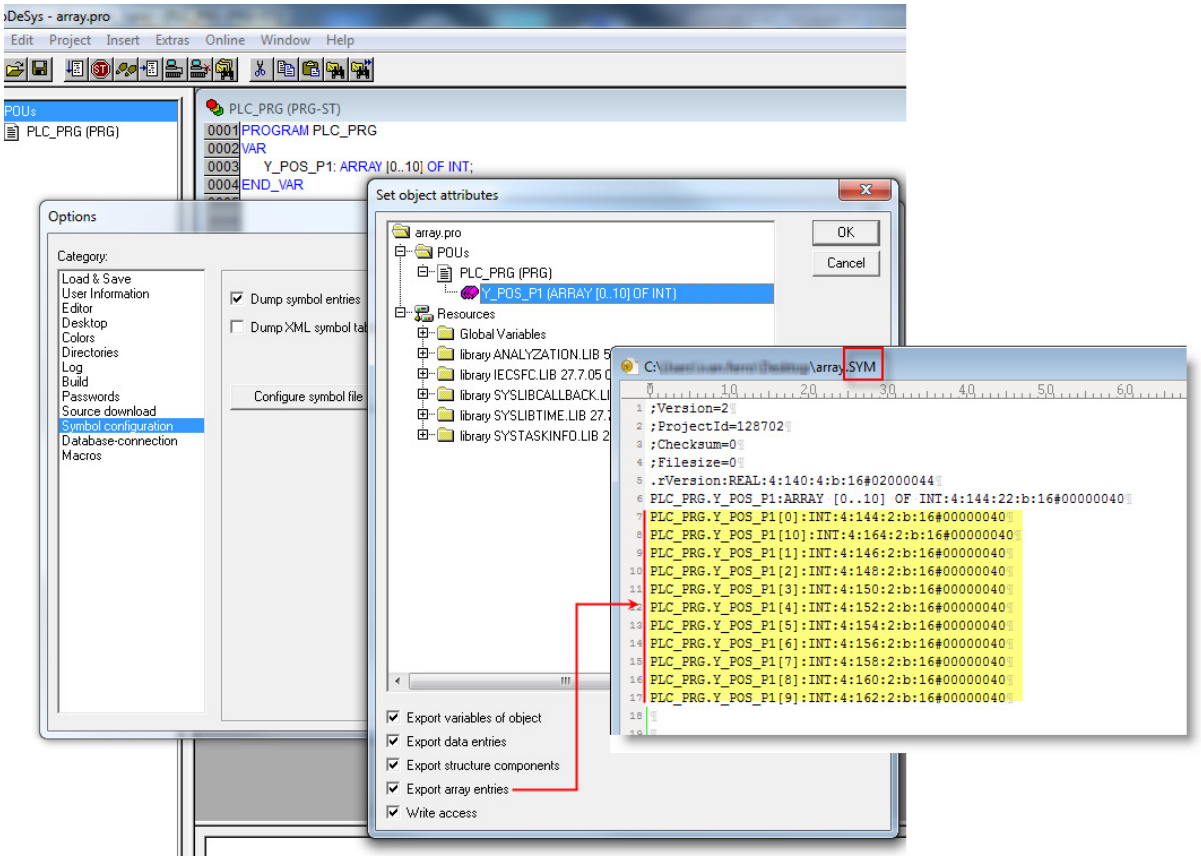


Figure 9

Note: When “Export array entries” is set, a tag for each element is created and exported into the SYM file. The entire tag list is automatically imported into Tag Editor.

The amount of tags can be reduced and only one Tag for each one array can be created by removing the checkbox “Export array entries”, see figure below.

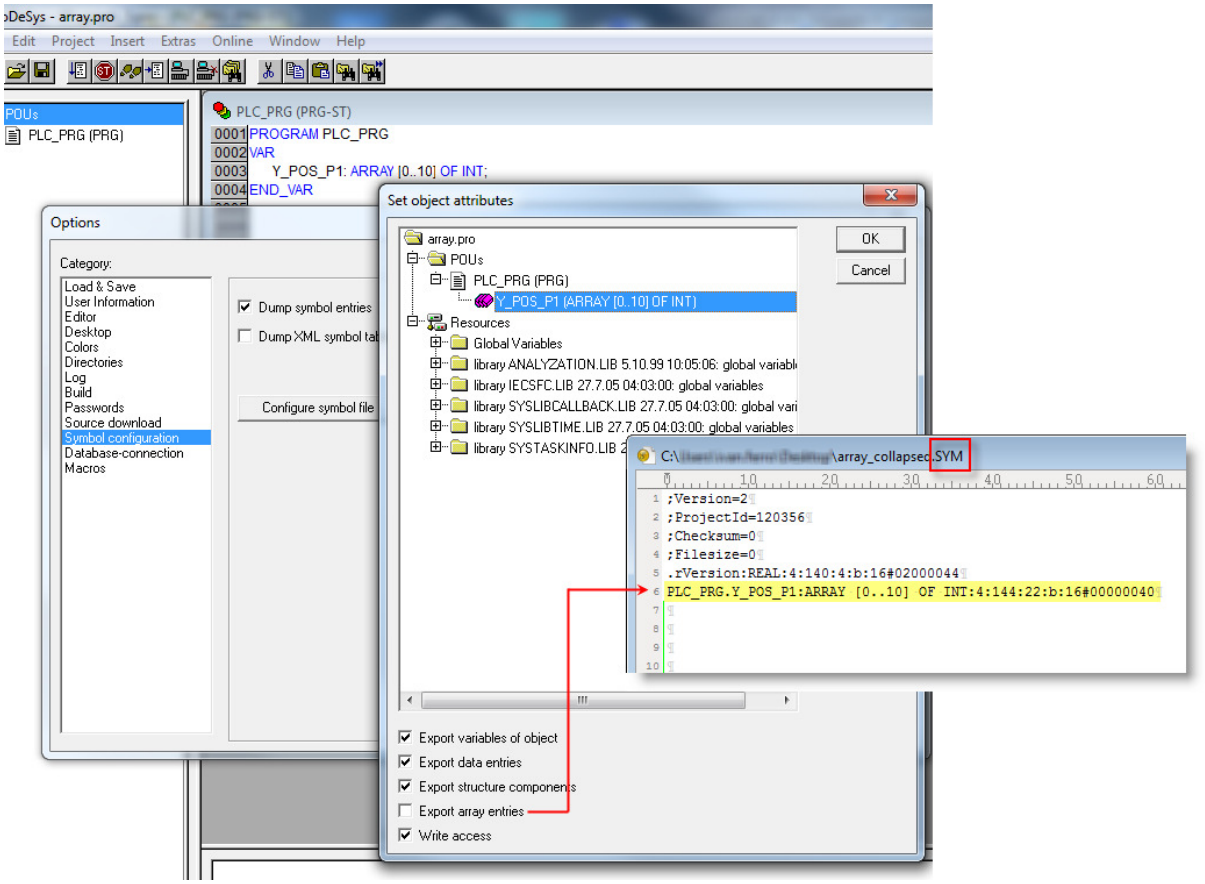
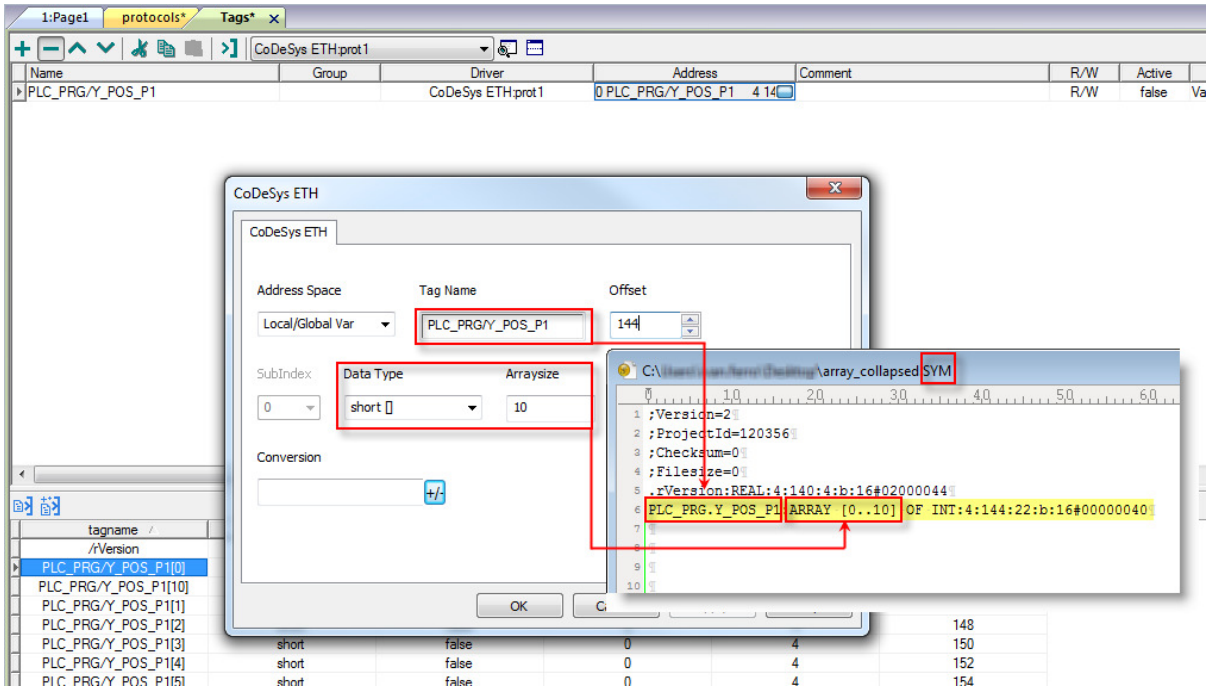


Figure 10

Note: When “Export array entries” is not set, only one tag is created and exported into the SYM file. The Array will not be automatically imported in Tag Editor and Tags need to be manually configured in Tag Editor



All Tag elements can be referenced in the editor using “TagIndex” in the “Attach to Tag” dialog

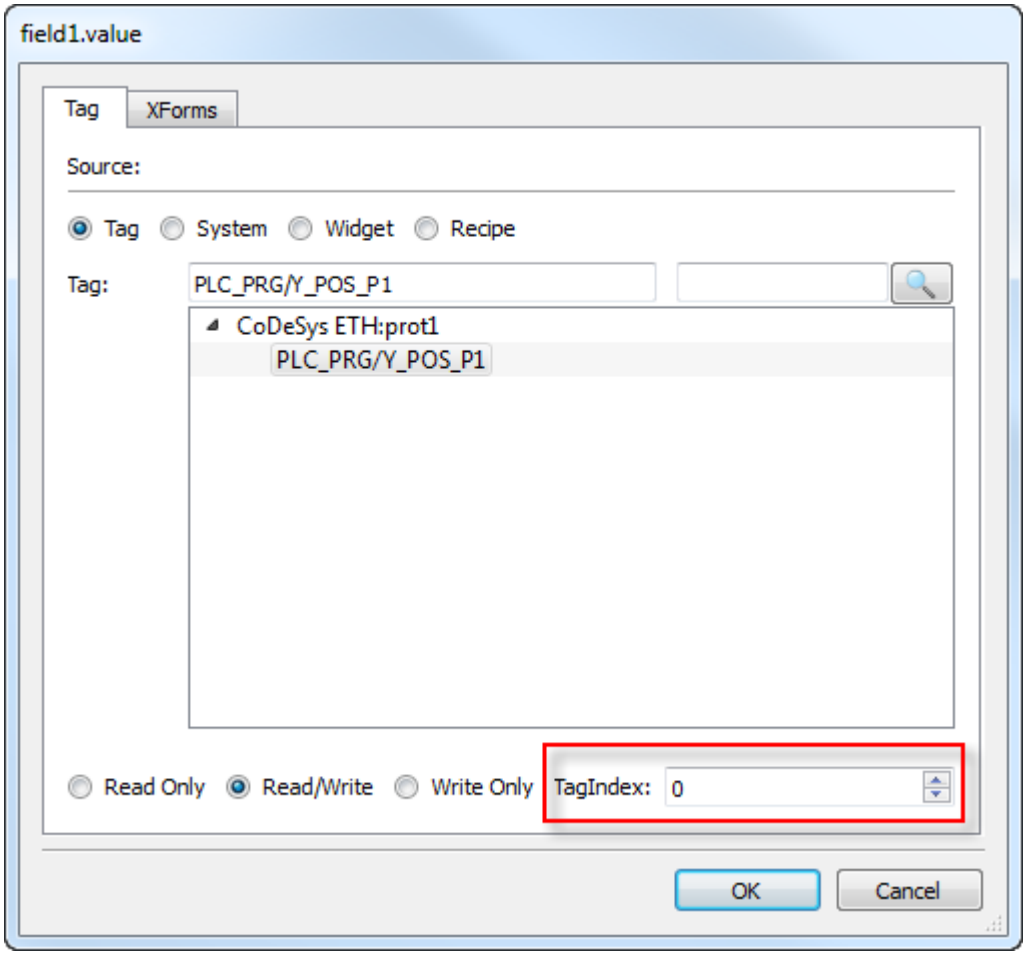


Figure 11

Data Types

The import module supports variables of standard data types and user defined data types.

Standard Data Types

The following data types in the CODESYS programming tool are considered standard data types by the import module:

BOOL
WORD
DWORD
INT
UINT
UDINT
DINT
STRING

REAL
TIME
DATE & TIME

and 1-dimensional ARRAY of the types above.

The 64-bit data types LWORD, LINT and LREAL are not supported.
String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35)) or default size (str: STRING) which is 80 characters.

Special Data Types

The CODESYS V2 ETH driver provides one special data type called "IP Override".

The IP Override IP allows changing at run time the IP address of the target controller you want to connect. This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.
The node override IP is initialized with the value of the controller IP specified in the project at programming time.
If the IP Override is set to 0.0.0.0, all the communication with the node is stopped, no request frames are generated anymore.
If the IP Override has a value different from 0.0.0.0, it is interpreted as node IP override and the target IP address is replaced run-time with the new value.
In case the panel has been configured to access to a network of controllers, each node has its own Override variable.

Note: the IP Override values assigned at run-time are retained through power cycles

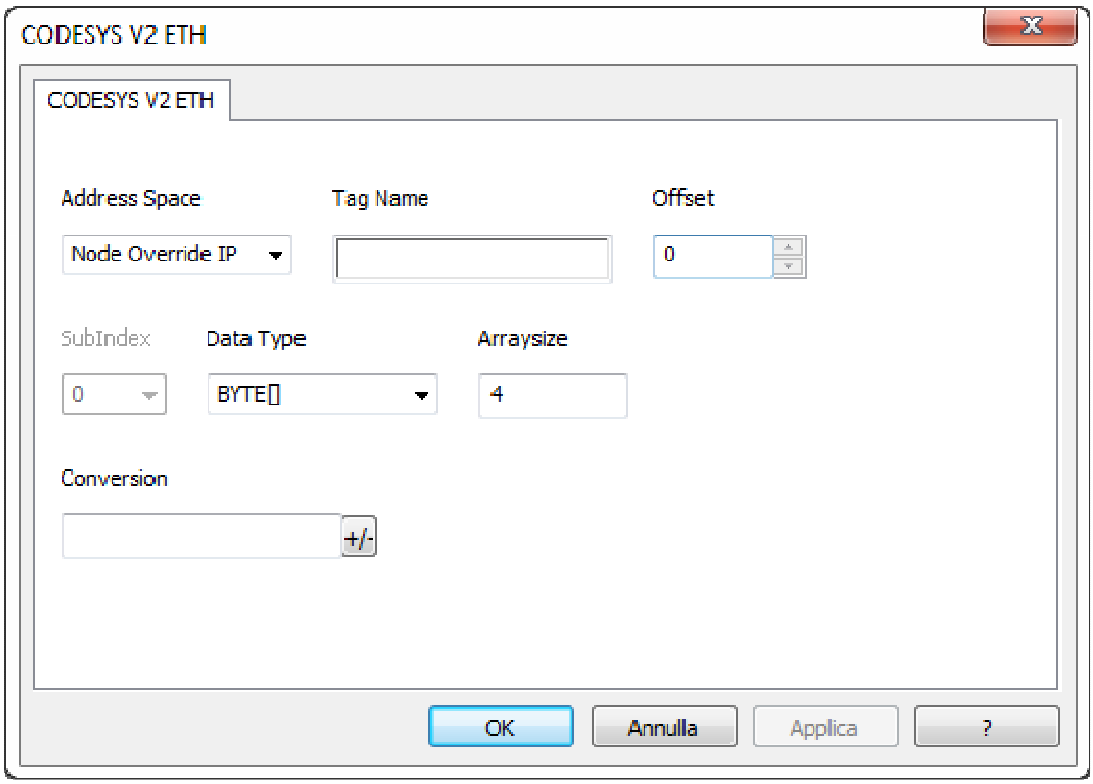


Figure 12

Limitations

Max block size is 1024 byte.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
Symbols file not present	Check Symbol file and download again the PLC program
“tag”not present in Symbols files	Check if the Tag is present into the PLC project
Time out on Acknoledge	
Time out on last Acknoledge	Controller didn’t sent last ack
Time out on data reciving	Controlled does not reply with data
Connection timeout	Device not connected